Details of `potMax` Analysis

Adam L. Pintar Dat Duthinh Emil Simiu

1 Introduction

The package `potMax` provides an estimation procedure for the distribution of the peak (maximum) value of a stationary, but otherwise fairly general, time series. It may also be used to calculate return values. The model that forms the basis of the procedure is a two dimensional Poisson process that is appropriate, in an asymptotic sense, for modeling the extremes of a random process. One dimension of the Poisson process is time since the target data sets are time series. An example of the second dimension is the pressure exerted by wind on a scale model of a building in a wind tunnel. The motivation for this R package are such time series, and the all sets provided with the package are examples of wind tunnel data. This vignette has three goals:

1. Describe the Poisson process to be used, including justification for its application to general random processes.
2. Provide the basic steps for estimation and uncertainty quantification in general terms, i.e., not specific to R [R, 2018] or this package, `potMax`.
3. Show how the package `potMax` may be used to carry out those steps.

2 Poisson Process

Poisson processes are defined by their intensity function. The two intensity functions used here are given in Equations 1 and 2.

$$\lambda(t, y) = \frac{1}{\sigma}\left[1 + \frac{k(y-\mu)}{\sigma}\right]_+^{-1/k-1} \tag{1}$$

$$\lambda(t, y) = \frac{1}{\sigma}\exp\left\{\frac{-(y-\mu)}{\sigma}\right\} \tag{2}$$

Notice that the left side of Equations 1 and 2 are functions of t (time) and y (e.g., pressure). However, the right side of Equations 1 and 2 are only functions of y. The implication is that only stationary time series, those not changing behavior over time, are considered. The $+$ subscript in Equation 1 means that

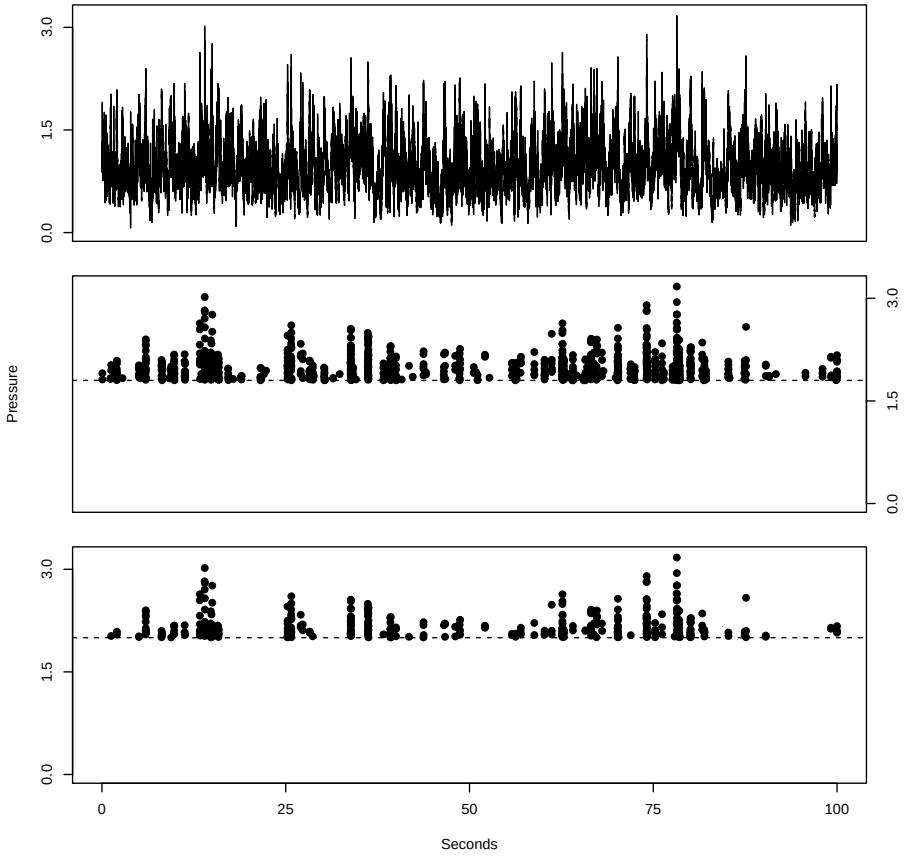


Figure 1: Raw time series (top), observations above 1.8 (middle), and observations above 2 (bottom).

negative values inside the square brackets are raised to zero. Equation 2 is the limit of Equation 1 as k approaches zero. For that reason, the Poisson process defined by the intensity function in Equation 2 is referred to as the Gumbel model henceforth. The two dimensional Poisson processes defined by Equations 1 and 2 are appropriate models for crossings of a high threshold. Consider Figure 1, which depicts a raw time series and two different thresholded versions. Notice in Figure 1 the "silos" of thresholded crossings. This occurs because the observations comprising the raw time series are autocorrelated. Figure 2 depicts an estimate of the autocorrelation function (ACF) for the series shown in the top plot of Figure 1. Observations separated by more than 40 increments of time (in this case almost one tenth of a second) remain highly positively correlated.

Poisson process are not appropriate for such data without further processing

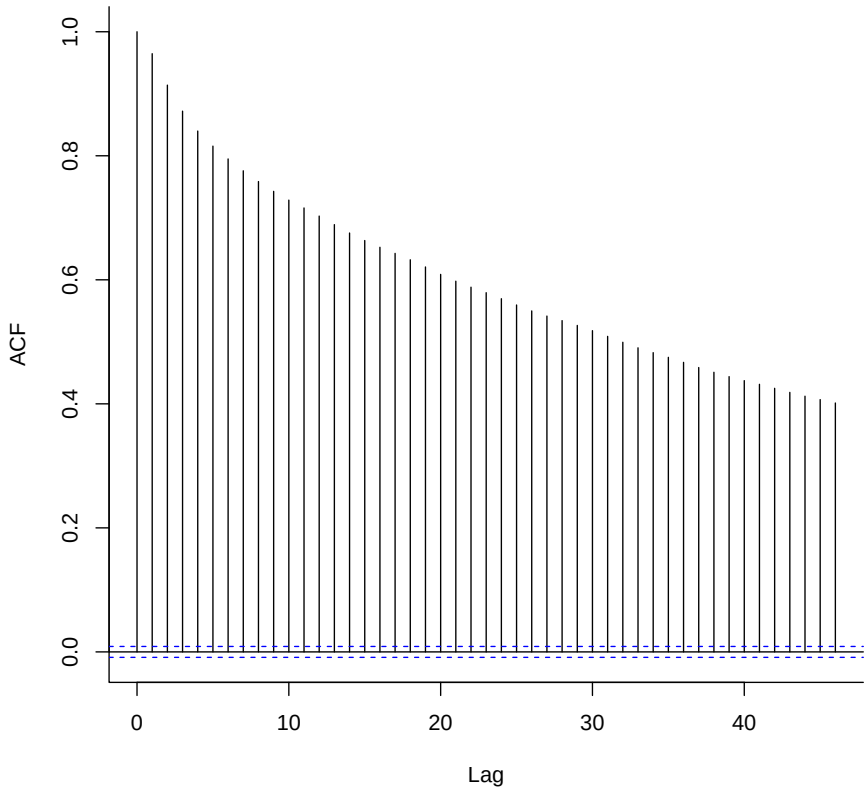


Figure 2: Estimated autocorrelation function for the time series in the top plot of Figure 1.

because one of their defining assumptions is independence. I am deliberately being vague because a careful treatment of Poisson processes is out of the scope of this vignette. So that the Poisson process model is tenable, the time series must be declustered before thresholding. Declustering proceeds by forming clusters and discarding all but the cluster maximums. Clusters are formed by groups of sequential observations falling above the series mean value. All observations below the series mean are ignored since the focus is on estimating the distribution of the peak value. Figure 3 displays the analog of Figure 1 after declustering, and Figure 4 depicts the estimated autocorrelation function of the series in the top plot of Figure 3. Figure 4 shows that the declustering is very effective for removing the autocorrelation. After removing the autocorrelation in the series, or declustering, the use of the Poisson processes defined by the intensity functions in Equations 1 and 2 as models for crossings of a high threshold is justified. They are used for such purposes in many papers, e.g., [Smith, 1989, Smith, 2004, Coles, 2004, Pintar et al., 2015, Duthinh et al., 2017]. The original theoretical justification is provided by [Pickands, 1971].

2.1 Threshold Choice

The intensity functions 1 and 2 are considered over the domain $\mathcal{D} = (0, \infty) \times (b, \infty)$, where b is a chosen threshold. A hurdle to the use of these models is the appropriate choice of b. Since the threshold dictates the data that are included in (or omitted from) fitting the model, its impact on the results can be large. Theory commands [Pickands, 1971] that the model becomes more appropriate as the threshold increases. However, since observations are taken over a finite period of time, the threshold cannot be chosen too high because too few observations will remain for fitting the model. Any approach to choosing a threshold must balance these competing aspects. A common and easy to implement approach is to pick a high quantile of the series, e.g., 95% (see page 489 of [Mannshardt et al., 2010]). This R package considers two alternatives.

2.1.1 Optimal Threshold

An optimal threshold based on the fit of the model to the data is the first alternative. The fit of the model is judged by the W-statistics defined in Equation (1.30) of [Smith, 2004]. Figure 5 shows a plot of W-statistics versus quantiles of the standard exponential distribution using the optimal threshold for the series in the top plot of Figure 3. If the data fit perfectly to the model, the points would fall exactly on the diagonal line. The threshold is chosen by creating such a plot for a sequence of potential thresholds and selecting the one that minimizes the maximum absolute vertical distance to the diagonal line. This approach is used in [Duthinh et al., 2017].

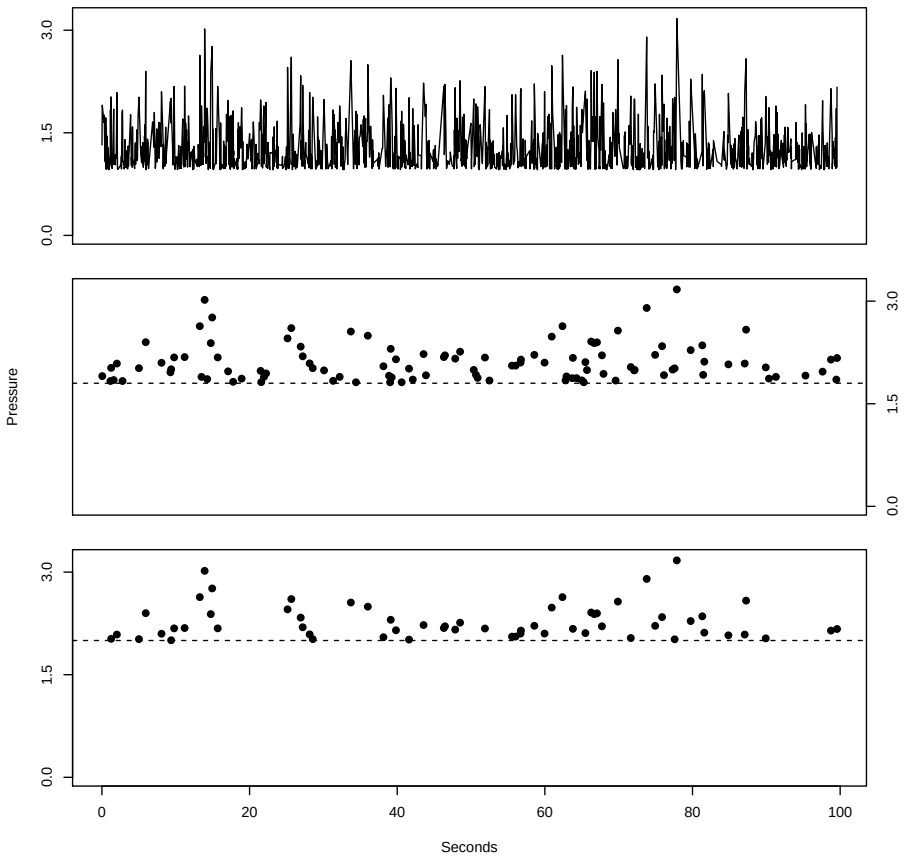


Figure 3: Raw time series (top), observations above 1.8 after declustering (middle), and observations above 2 after declustering (bottom).

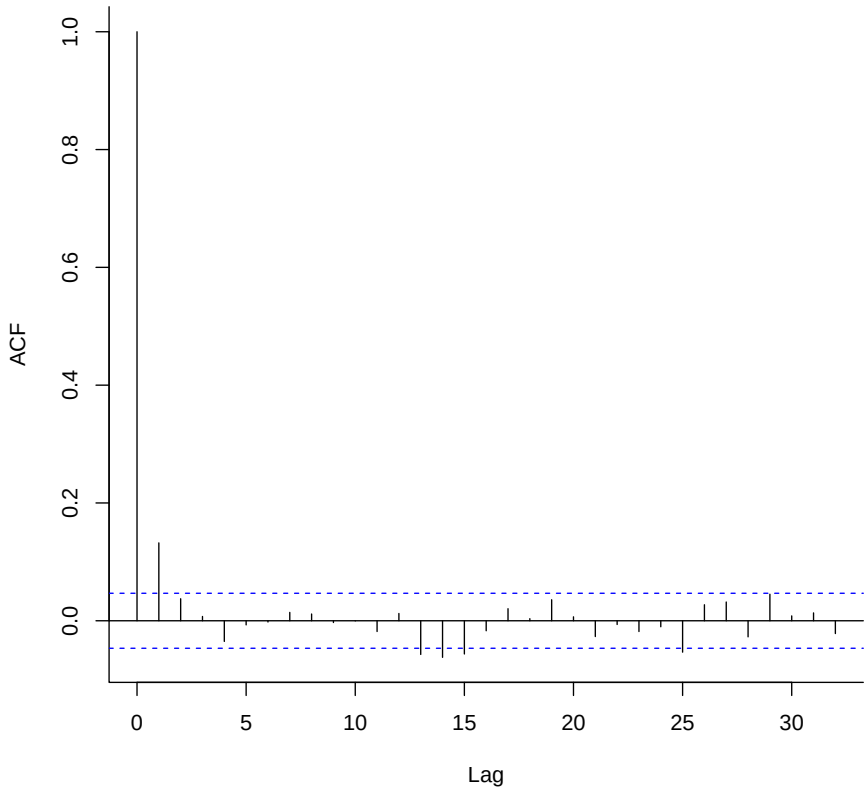


Figure 4: Estimated autocorrelation function for the time series in the top plot of Figure 1 after declustering it.

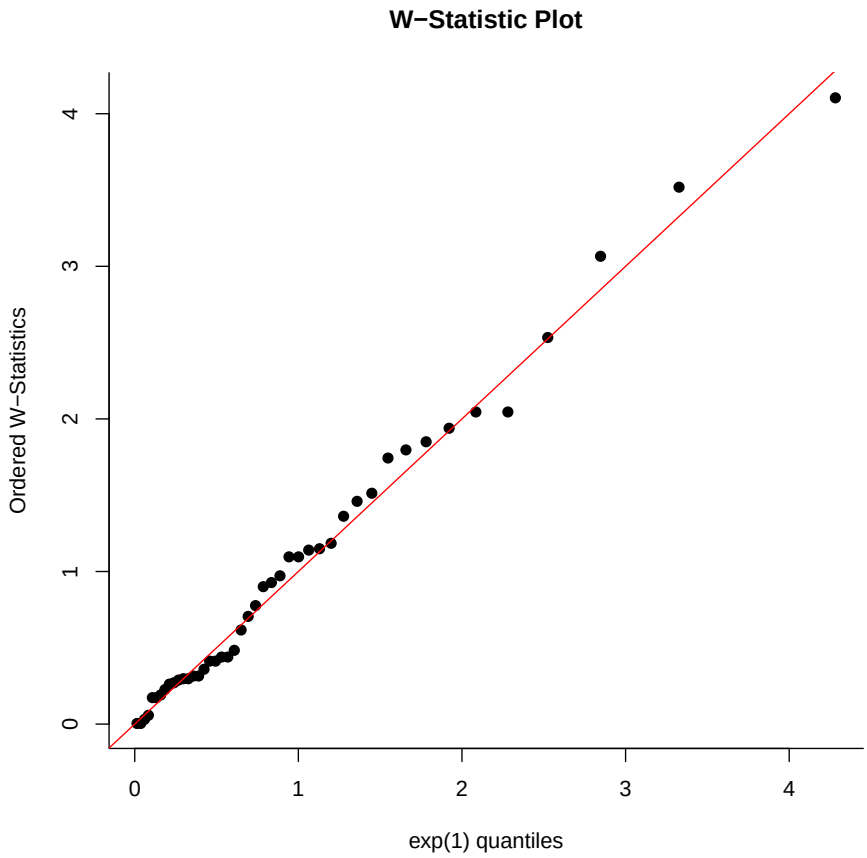


Figure 5: Plot of the W-statistics versus their corresponding standard exponential quantiles for the declustered series depicted in the top plot of 3 using the optimal threshold.

2.1.2 Many Thresholds

The second alternative is to combine results for many thresholds by weighted averaging. The key step then is constructing the weights, for which the W-statistics may again be leveraged. Let δ_j be the maximum absolute vertical distance in a plot of the W-statistics against the standard exponential quantiles, normalized to the unit interval for the threshold b_j. A natural transformation of the δ_j is

$$w_j = \frac{\exp\{-\tau\delta_j\}}{\sum_j \exp\{-\tau\delta_j\}} \tag{3}$$

With $\tau = 0$, each threshold recieves equal weight, and as τ approaches ∞, the weight corresponding to the smallest δ_j approaches unity while the rest approach zero. A reasonable setting seems to be $\tau = 5$ [Pintar and Duthinh, in prep].

2.2 Estimation

The model parameters, η=(μ, σ, k) for the intensity function in Equation 1, and η=(μ, σ) for the intensity function in Equation 2 are estimated by maximum likelihood. Let (y_i, t_i) be the declustered values that lie above the threshold b. The likelihood is then given by Equation 4.

$$L(\eta) = \left(\prod_{i=1}^{I} \lambda(t_i, y_i)\right) \cdot \exp\left\{-\int_{\mathcal{D}} \lambda(t, y)dtdy\right\} \tag{4}$$

The domain of integration, $\mathcal{D}$, in Equation 4 is the unbounded rectangle $[0, T] \times [b, \infty)$, where T is the time of the last observation.

3 Estimating the Distribution of the Peak

The goal of this section is to describe the algorithm for estimating the distribution of the peak in general statistical terms, i.e., without reference to R or the `potMax` package. It is expressed as four steps, most with substeps of their own.

1. Decluster the series.
2. Select the threshold or thresholds
   (a) Construct a set of potential thresholds. This may be done by specifying a minimum and maximum number of observations and identifying their respective thresholds, say b_1 and b_B. Then, the thresholds are $b_1 > b_2 > \cdots > b_{B-1} > b_B$ such that if b_j corresponds to n_j observations, b_{j+1} corresponds to $n_j + 1$ observations (assuming no ties).
   (b) For each potential threshold, fit the model via maximum likelihood as described in Section 2.2.
   (c) Create a W-plot for each fit, as described in Section 2.1.1.

(d) Summarize each W-plot by the maximum absolute vertical distance from the points to the 45° line

(e) Select the threshold that minimizes the maximum distance as described in Section 2.1.1, or use the distances to specify a weight associated with each threshold as described in Section 2.1.2

3. Empirically build the distribution of the peak

   (a) Generate a series of desired length from the fitted model
   - This may be accomplished, for example, by algorithm 9 of [Pasupathy, 2011]
   - If results of many thresholds are being combined, select a threshold at random from the collection according to the weights, i.e., threholds with high weights should be selected more often.

   (b) Record the peak of the generated series

   (c) Repeat (a) and (b) n_{mc} times; the recorded peaks form an empirical approximation to the distribution of the peak, or mixture of peak distributions in the case of many thresholds.

4. Quantify uncertainty

   - For a single optimal threhold
     (a) Sample n_{boot} values from a multivariate Gaussian distribution with mean equal to the estimated parameters, $\widehat{\eta}$, and covariance matrix equal to the negative inverse Hessian matrix of the log-likilihood evaluated at its maximum
     (b) For each set of sampled parameters reapeat step 3
   - For a collection of thresholds
     (a) Sample the declustered data with replacement (a bootstrap sample)
     (b) For each potential threshold refit the model
     (c) Recalculate the weights
     (d) Repeat step 3
     (e) Repeat (a) – (d) *n_boot* times

The result of step 4 is n_{boot} empirical approximations to the distribution of the peak, or mixture of peak distributions in the case of many thresholds.

4 Return Values

Calculation of return values follows an algorithm similar to that in Section 5.3. The only difference is step 3. Instead of empirically building the peak distribution, the equation

$$\int_{y_N}^{\infty} \int_{0}^{1} \lambda(t, y) dt dy = \frac{1}{N} \tag{5}$$

is solved for y_N, which is interpreted as the N-year, -month, -week, etc. return value. In the case of many thresholds, Equation 5 is sloved for each threshold and the solutions are combined by weighted averaging according the weights calculated in step 2.

5 Using potMax

The potMax package is demonstrated on the data in the top plot of Figure 1. The data set is distributed as part of the potMax package and is referenced by the name jp1tap1715wind270. The present sections describe the functions to call to carryout the steps described in Section 5.3.

5.1 Declustering

```
complete_series <- -jp1tap1715wind270$value
declustered_obs <- decluster(complete_series = complete_series)
```

The argument complete_series is the time series itself. Note the negative sign because deep valleys are more interesting for this data set than high peaks. There exists a second optional argument to the function decluster. See the help file for more information.

5.2 Threshold Selection and Calculation of Weights

```
thresholded_obs <- gumbelEstThreshold(x = declustered_obs,
                                      lt = 100,
                                      n_min = 10,
                                      n_max = 100)
summary(thresholded_obs)

## Selected threshold:  2.10005
##
## Number of thresholded observations:  45
##
## Summary statistics:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.101   2.178   2.284   2.361   2.481   3.172

pot_fit <- gumbelMLE(x = thresholded_obs, hessian_tf = TRUE)
```

The first argument x is the output from the function decluster. The second argument, lt, is the length of observation time for the series, in this case 100 seconds. The unit for time can be anything. It is the burden of the user to be consistent and interpret results according to the supplied unit. The third

argument, `n_min`, specifies the minimum number of observations to be allowed after thresholding. This argument affects how many potential thresholds are considered in the search for an optimal threshold. For the Gumbel model, I recommend 10 or more, and for the full model, I recommend 15 or more. The fourth argument, `n_max`, complements `n_min`, specifying the maximum number of observations to allow after thresholding. This should be chosen sufficiently high that the selected threshold does not correspond to that number of observations, but not so high that run times become unbearable. Note that the output of `gumbelEstThreshold` must be passed into `gumbelMLE` once more so that the Hessian matrix may be calculated, which is not done by `gumbelEstThreshold`. This is necessary for uncertainty quantification.

The W-plot for the fitted model can be creted with the function `gumbelWPlot`. See the help file for `gumbelWplot` for a description of its arguments. The plot for the example series is found in Figure 5.

A plot of all proposed thresholds against the corresponding fit statistic is available too, Figure 6.

```
threshPlot(thresholded_obs)
```

If it is desired to combine the results from multiple thresholds, the function `gumbelMultiFit` is called in place of both, `gumbelEstThreshold` and `gumbelMLE`. The required arguments are the same as for `gumbelEstThreshold`, except the additional `weight_scale`, for which a good choice is 5, as described in Section 2.1.2

```
multi_pot_fit <- gumbelMultiFit(x = declustered_obs, lt = 100,
                                n_min = 10, n_max = 100,
                                weight_scale = 5)
summary(multi_pot_fit)
```

```
##          mu     sigma  thresh       weight
## 1  2.022685 0.2182393 2.52520 0.0048787371
## 2  1.974488 0.2327130 2.48815 0.0105928637
## 3  1.975614 0.2324410 2.46845 0.0103464473
## 4  1.923555 0.2494802 2.43255 0.0265174776
## 5  1.897873 0.2578066 2.40475 0.0330347072
## 6  1.932051 0.2461094 2.39895 0.0210195406
## 7  1.956568 0.2376056 2.39200 0.0117552190
## 8  1.940139 0.2434056 2.38620 0.0176414742
## 9  1.971206 0.2325604 2.37000 0.0081044460
## 10 1.946987 0.2414678 2.34800 0.0159588816
## 11 1.955003 0.2384044 2.33870 0.0124253458
## 12 1.934167 0.2465851 2.31900 0.0238594563
## 13 1.899740 0.2600904 2.29355 0.0252442340
## 14 1.877019 0.2692300 2.27270 0.0213700903
## 15 1.834790 0.2865288 2.24370 0.0048537870
```

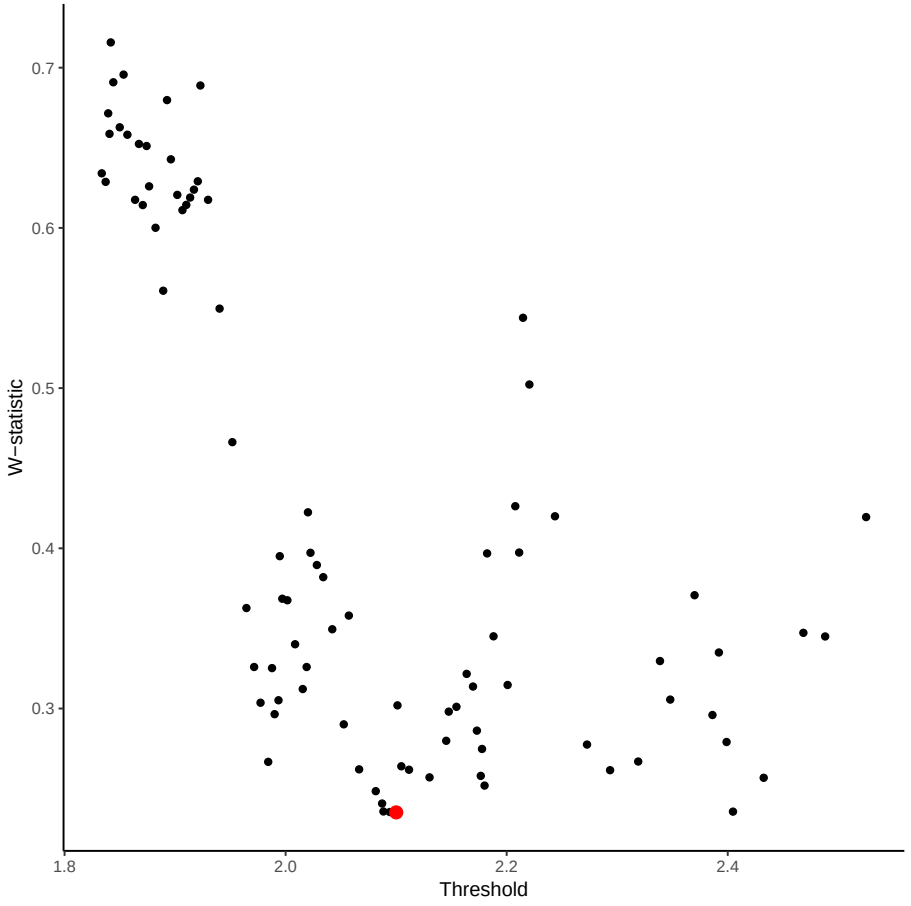


Figure 6: Plot of fit statistics versus thresholds. The red point identifies the minimum.

```
## 16 1.808097 0.2975217 2.22055 0.0020627226
## 17 1.794376 0.3032718 2.21480 0.0013374297
## 18 1.839047 0.2843070 2.21130 0.0061423600
## 19 1.830965 0.2878070 2.20780 0.0045466403
## 20 1.860548 0.2749082 2.20085 0.0145155764
## 21 1.853079 0.2782628 2.18810 0.0105834995
## 22 1.840296 0.2840629 2.18230 0.0061775826
## 23 1.873938 0.2686089 2.18000 0.0278967903
## 24 1.869017 0.2709089 2.17770 0.0219940675
## 25 1.900242 0.2561230 2.17655 0.0261932067
## 26 1.908209 0.2522723 2.17305 0.0195252190
## 27 1.915466 0.2486994 2.16955 0.0146632295
## 28 1.917527 0.2476967 2.16380 0.0135064105
## 29 1.912357 0.2503073 2.15455 0.0167270581
## 30 1.911520 0.2507199 2.14760 0.0172559585
## 31 1.907054 0.2530200 2.14530 0.0208546704
## 32 1.902149 0.2557775 2.13020 0.0264490925
## 33 1.879266 0.2678781 2.11165 0.0251688505
## 34 1.878079 0.2685181 2.10470 0.0246098648
## 35 1.871625 0.2720182 2.10120 0.0165719192
## 36 1.891577 0.2610784 2.10005 0.0332029202
## 37 1.891527 0.2611278 2.09430 0.0331250557
## 38 1.891232 0.2612740 2.08850 0.0330049816
## 39 1.889126 0.2624740 2.08730 0.0313400804
## 40 1.897855 0.2575107 2.08155 0.0289257202
## 41 1.881225 0.2672955 2.06650 0.0251057568
## 42 1.865479 0.2765956 2.05720 0.0092489764
## 43 1.875675 0.2705572 2.05260 0.0187431796
## 44 1.867122 0.2757656 2.04220 0.0101075554
## 45 1.862318 0.2787014 2.03405 0.0072076121
## 46 1.861220 0.2793913 2.02825 0.0066606664
## 47 1.860098 0.2800906 2.02250 0.0061526949
## 48 1.856465 0.2823906 2.02020 0.0047303096
## 49 1.869825 0.2738530 2.01900 0.0129168684
## 50 1.871619 0.2726911 2.01550 0.0149028369
## 51 1.868043 0.2750579 2.00855 0.0111425419
## 52 1.864471 0.2774225 2.00160 0.0083746393
## 53 1.864340 0.2775108 1.99700 0.0082913041
## 54 1.860940 0.2798108 1.99470 0.0062887617
## 55 1.872063 0.2722168 1.99355 0.0160296030
## 56 1.873088 0.2715111 1.99005 0.0175470228
## 57 1.869725 0.2738611 1.98770 0.0130092777
## 58 1.876467 0.2691361 1.98425 0.0239150375
## 59 1.872370 0.2720775 1.97730 0.0162892944
## 60 1.869872 0.2738836 1.97150 0.0129144334
```

```
## 61 1.865791 0.2768881 1.96455 0.0088081854
## 62 1.853960 0.2856728 1.95180 0.0029991136
## 63 1.843951 0.2931427 1.94025 0.0012601792
## 64 1.835600 0.2994799 1.92985 0.0006217707
## 65 1.826463 0.3064299 1.92290 0.0002959408
## 66 1.834070 0.3006084 1.92055 0.0005512777
## 67 1.834685 0.3001221 1.91705 0.0005819455
## 68 1.835280 0.2996595 1.91360 0.0006126170
## 69 1.835801 0.2992382 1.91015 0.0006424979
## 70 1.836184 0.2989358 1.90665 0.0006647578
## 71 1.835097 0.2998197 1.90200 0.0006021661
## 72 1.832585 0.3018899 1.89620 0.0004779378
## 73 1.828348 0.3053899 1.89270 0.0003253779
## 74 1.841393 0.2944685 1.88925 0.0011218760
## 75 1.837362 0.2979544 1.88230 0.0007455064
## 76 1.834681 0.3002894 1.87650 0.0005696512
## 77 1.832061 0.3025894 1.87420 0.0004383643
## 78 1.835878 0.2992396 1.87075 0.0006432628
## 79 1.832026 0.3026896 1.86730 0.0004325293
## 80 1.835550 0.2995371 1.86380 0.0006217061
## 81 1.831569 0.3031932 1.85685 0.0004074074
## 82 1.827832 0.3066433 1.85340 0.0002756955
## 83 1.831113 0.3036189 1.84990 0.0003882179
## 84 1.828395 0.3061861 1.84410 0.0002897807
## 85 1.825977 0.3084861 1.84180 0.0002237195
## 86 1.831412 0.3032755 1.84065 0.0004051140
## 87 1.830227 0.3044255 1.83950 0.0003546016
## 88 1.834129 0.3006255 1.83715 0.0005533402
## 89 1.833650 0.3010957 1.83365 0.0005234730
```

5.3 Estimation of the Distribution of the Peak

The call to `gumbelMaxDist` is the same for a single optimal threshold and many thresholds.

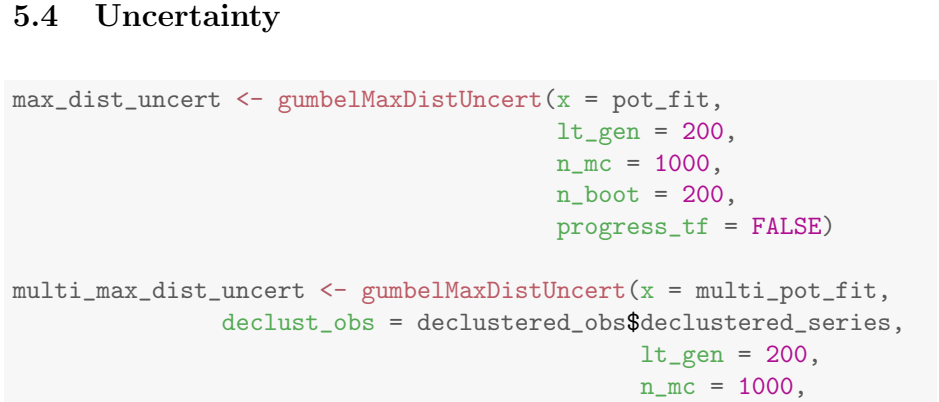
```
max_dist <- gumbelMaxDist(x = pot_fit,
                          lt_gen = 200,
                          n_mc = 1000,
                          progress_tf = FALSE)

multi_max_dist <- gumbelMaxDist(x = multi_pot_fit,
                                lt_gen = 200,
                                n_mc = 1000,
                                progress_tf = FALSE)
```

The argument `lt_gen` provides the length of the series for which the distibutio of the maximum is sought, which could be different from the lenght of the original series, but with consistent units. The argument `n_mc` is the number of samples to draw from the distribution of the maximum. A progress bar may optionally be drawn.

For a single optimal threshold as well as many thresholds, the mean of the distribution may be calculated with the function `mean`, and the entire distribution may be plotted as a histogram with an S3 method for the generic `plot` function.

```
mean(max_dist)

## [1] 3.41819

plot(max_dist)
```

```
mean(multi_max_dist)

## [1] 3.43593

plot(multi_max_dist)
```



5.4 Uncertainty

```
max_dist_uncert <- gumbelMaxDistUncert(x = pot_fit,
                                       lt_gen = 200,
                                       n_mc = 1000,
                                       n_boot = 200,
                                       progress_tf = FALSE)

multi_max_dist_uncert <- gumbelMaxDistUncert(x = multi_pot_fit,
                declust_obs = declustered_obs$declustered_series,
                                             lt_gen = 200,
                                             n_mc = 1000,
                                             n_boot = 200,
                                             progress_tf = FALSE)
```

The argument `n_boot` corresponds to the number of bootstrap replicates of the peak distribution to calculate. Note that the `lt_gen` argument in the call to `gumbelMaxDistUncert` should match the `lt_gen` argument provided in the call to `gumbelMaxDist`. All other arguments are as previously described. For the `gumbel_multi_fit` S3 method, the original declustered observations must be passed in as an argument so that boostrap samples may be constructed.

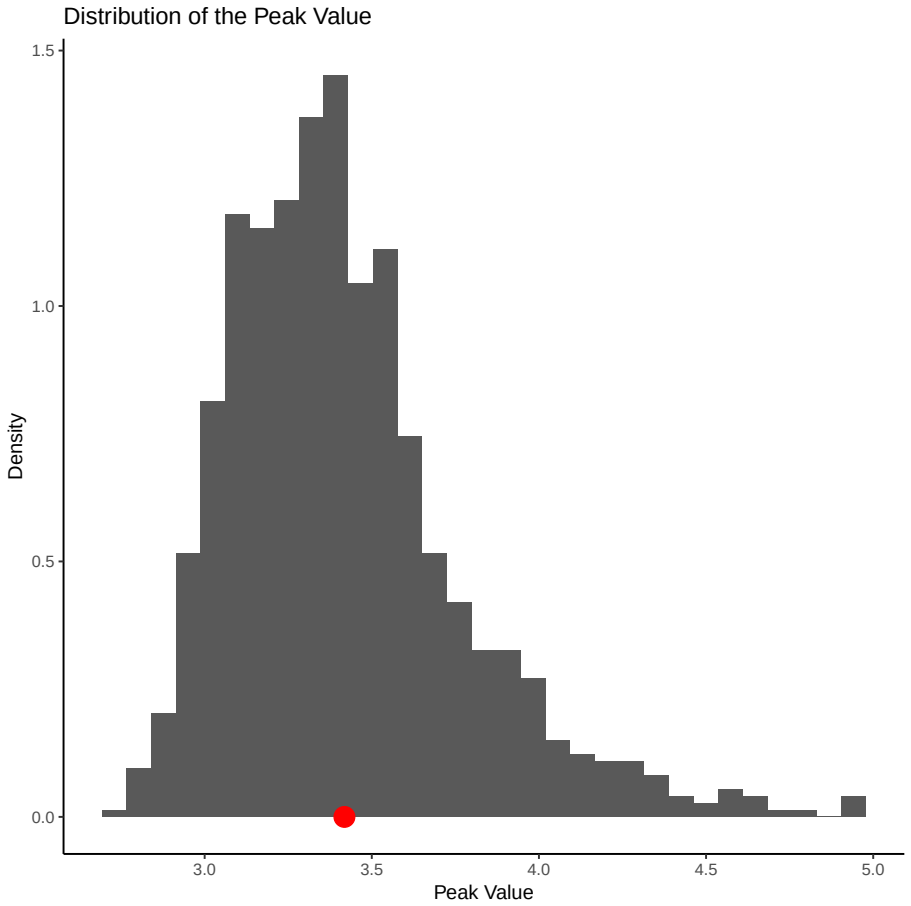


Figure 7: Histogram of the estimated distribution of the peak value considering only a single optimal threshold. The red point shows the mean of the distribution.

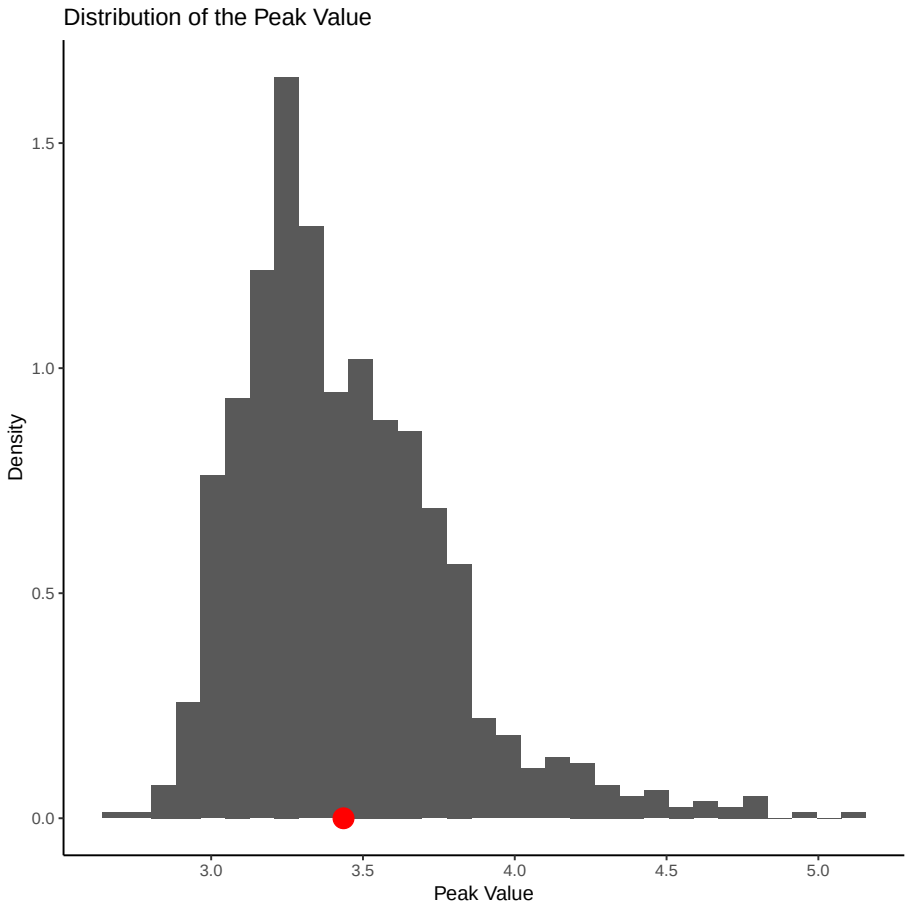


Figure 8: Histogram of the estimated distribution of the peak value combining results from many thresholds. The red point shows the mean of the distribution.

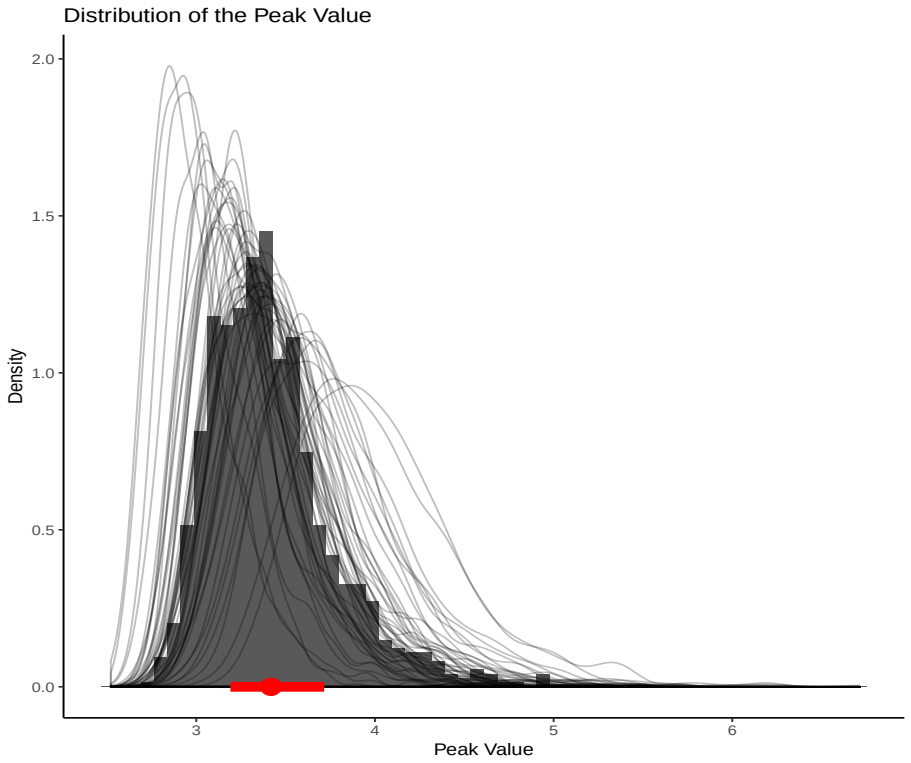


Figure 9: Same as Figure 7, but adding bootstrap replictes of the distribution of the peak value to convey uncertainty in estimation. The red line shows an 80% confidence interval for the mean of the distribution of the peak.

The results maybe added to the histogram depicting the distribution of the peak (Figure 10),or they may be displayed alone (Figure 10). The same `plot` commands work for the S3 classes `gumbel_max_dist_multi_thresh` and `gumbel_max_dist_uncert_multi_thresh`.

```
# Figure 9
plot(max_dist_uncert, add = TRUE)
```

```
# Figure 10
plot(max_dist_uncert, add = FALSE)
```

5.5 Return Values

The calls to `gumbelMaxDist` and `gumbelMaxDistUncert` are replaced by `gumbelNYear` and `gumbelNYearUncert`. Plot and summary functions are not always imple-

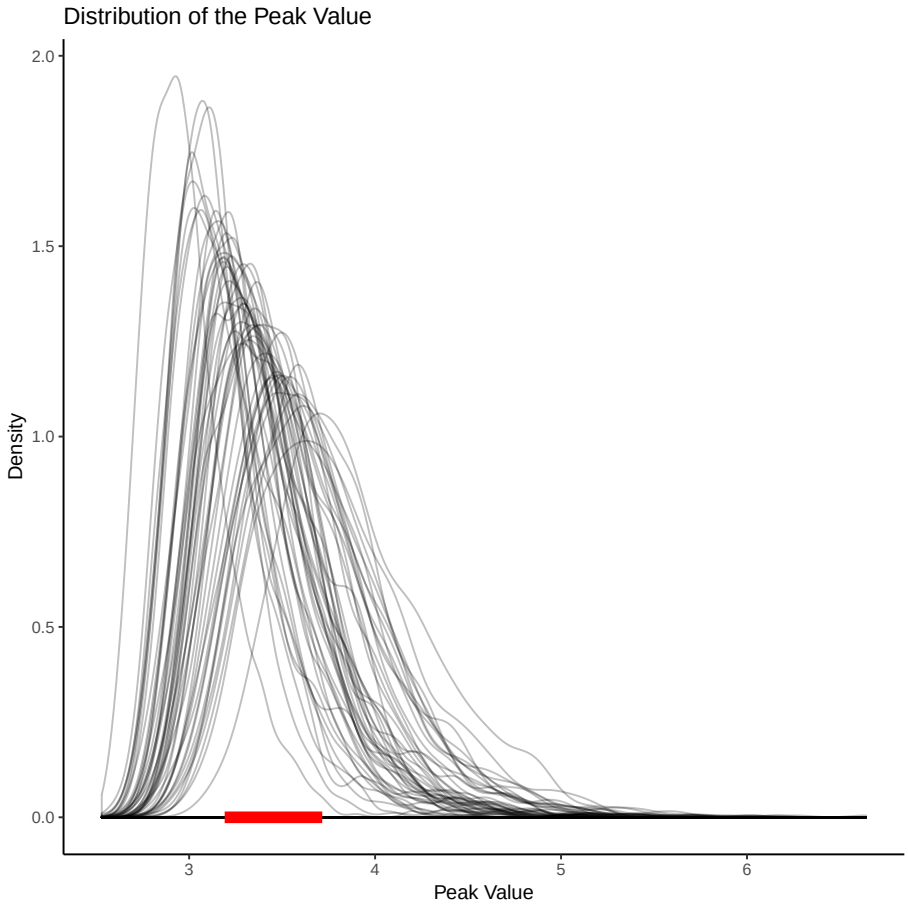


Figure 10: Bootstrap replicates of the distribution of the peak starting with the series shown in the top plot of Figure 1. The red line shows an 80% confidence interval for the mean of the distribution of the peak value.

mented for return values.

```
return_200s <- gumbelNYear(x = pot_fit, N = 200)
return_200s$N_year_val

## [1] 3.274853

return_200s_uncert <- gumbelNYearUncert(x = pot_fit, N = 200,
                                        n_boot = 200)
quantile(return_200s_uncert$boot_samps, probs = c(0.1, 0.9))

##      10%      90%
## 3.061506 3.515152

multi_return_200s <- gumbelNYear(x = multi_pot_fit, N = 200)
multi_return_200s$N_year_val

## [1] 3.282876

multi_return_200s_uncert <- gumbelNYearUncert(x = multi_pot_fit,
               declust_obs = declustered_obs$declustered_series,
                                              N = 200,
                                              n_boot = 200)
summary(multi_return_200s_uncert)

## $standard_error
## [1] 0.1545509
##
## $conf_int
##      10%      90%
## 3.076580 3.460663
```

6 Full Estimation

The steps shown in Section 5 leveraged the intensity function in Equation 2. To use the intensity function in Equation 1 instead, replace the function calls `gumbel*` with `full*`. For example, to choose an optimal threshold, use the function `fullEstThreshold`. Note that the functions `fullEstThreshold`, `fullMLE`, and `fullMultiFit` have an additional required argument, `n_starts`, which specifies the number of times to perform the optimization from random starting locations. The method for `fullMaxDistUncert` for the S3 class `full_multi_fit` also requires this argument. A reasonable value is 20. *Note that functions associated with return values are only implemented for the the Gumbel model, i.e. the intensity function in Equation 2.*

References

[Coles, 2004] Coles, S., "The use and Misuse of Extreme Value Models in Practice," in *Extreme Values in Finance, Telecomunications, and the Environment*, Finkenstädt, B. and Rootzén, H., editors, chapter 2, 79–100, Chapman & Hall/CRC (2004).

[Duthinh et al., 2017] Duthinh, D., Pintar, A.L., and Simiu, E. "Estimating peaks of stationary random processes: A peaks-over-threshold approach," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering* **3**(4), URL `https://doi.org/10.1061/AJRUA6.0000933` (2017).

[Mannshardt et al., 2010] Mannshardt-Shamseldin, E. C., Smith, R. L., Sain, S. R., Mearns, L. O., and Cooley, D., "Downscaling Extremes: A Comparison of Extremem Value Distributions in Point-source and gridded Precipitation Data," *The Annals of Applied Statistics*, **4**, 484–502 (2010).

[Nelder and Mead, 1965] Nelder, J. A. and Mead, R., "A Simplex Method for Function Minimization," *The Computer Journal*, **7**, 308–313 (1965).

[Pasupathy, 2011] Pasupathy, R., "Generating nonhomogeneous Poisson processes." In *Wiley encyclopedia of operations research and management science* , Hoboken, NJ: Wiley (2011).

[Pickands, 1971] Pickands, J. III, "The Two-dimensional Poisson Process and Extremal Processes," *Journal of Applied Probability*, **8**, 745–756 (1971).

[Pintar et al., 2015] Pintar, A. L., Simiu, E., Lombardo, F. T., and Levitan, M., "Maps of Non-hurricane Non-tornadic Winds Speeds With Specified Mean Recurrence Intervals for the Contiguous United States Using a Two-dimensional Poisson Process Extreme Value Model and Local Regression," *NIST Special PUblication 500-301*, URL `http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-301.pdf` (2015).

[Pintar and Duthinh, in prep] Pintar, A. L. and Duthinh, D., "Threshold Uncertainty in Poisson Process Extreme Vaue Models," (in preparation).

[R, 2018] R Core Team, "R: A Language and Environment for Statistical Computing," R Foundation for Statistical Computing, Vienna, Austria, URL `https://www.R-project.org/` (2018).

[Smith, 1989] Smith, R. L., "Extreme Value Analysis of Environmental Time Series: an Application to Trend Detection in Ground-level Ozone," *Statistical Science*, **4**, 367–393 (1989).

[Smith, 2004] Smith, R. L., "Statistics of Extremes, with Applications in Environment, Insurance, and Finance," in *Extreme Values in Finance, Telecomunications, and the Environment*, Finkenstädt, B. and Rootzén, H., editors, chapter 1, 1–78, Chapman & Hall/CRC (2004).